

REMARKS

Claims 1-23 are pending in the present application. Claims 1, 12, and 18 are the independent claims. In the Official Action, dated September 16, 2003, claims 12-23 were rejected under 35 U.S.C. § 102(e) as allegedly anticipated by U.S. Patent No. 6,606,740 B1 ("Lynn et al."). Claims 1-11 were rejected under 35 U.S.C § 103(a) as being allegedly obvious over Lynn et al. in view of U.S. Patent No. 5,729,747 ("Tsukakoshi"). The outstanding rejections are respectfully traversed.

Summary of the Invention

There is often a disconnect between the language of the business people who envision software and the needs of the programmers who design and implement software. This disconnect may result in the software developers' failure to capture all of the "use cases" that may be helpful for users of software because the developers do not fully appreciate the nature of the business process that they will implement. **A "use case" is an instance of the use of the software by an actor.** Application 2, line 2.

The present invention provides a method and tool for designing software. Application 3, line 7-8. First, a business plan for the software is specified. Next, one or more "focus areas" are identified. Application 3, line 9-10. Each focus area includes a set of "requirements" for the system. Application 3, lines 10-11. These requirements include a description of a process to be performed and any constraints on the manner in which the process is to be performed. Application 3, lines 11-14. The focus area also includes a set of "participants" who will interact with the specified process. Application 3, lines 14-15. **Each participant may have one or more identifiable "roles."** Application 3, lines 15-16.

The focus area may be decomposed into several "sub" focus areas. Application 3, line 20. Sub-focus areas are created by identifying aspects of the original focus area. Application 3, line 21-22. Each sub-focus area has a set of one or more participants. Application 3, line 25-26. Focus areas are decomposed recursively into lower and lower levels until each of the sub-participants (i.e., actors) in the tasks covered by the lowest level focus area has only one role. Application 4, line 2-4. **A focus area where all of the participants have only one role is analogous to a "business use case," which may then be modeled by conventional means.** Application 4, lines 4-6.

Lynn et al.

Lynn et al. discloses a framework for developing a workflow processing system using object oriented design principles to minimize coding effort. *See* Lynn et al. column 2, lines 56-60. It provides a set of software objects, each uniquely performing a function in a workflow processing system. *See* Lynn et al. column 4, lines 52-54. The functionality provided by the objects is broad enough to apply to many different line of business applications. *See* Lynn et al. column 4, lines 64-66. Different applications and different users would utilize different subsets of the software objects to accomplish the work they are conducting. *See* Lynn et al. column 5, lines 62-65.

The software structure disclosed in Lynn et al. is shown in Lynn et al. figure 2. The corresponding text, found from column 5, lines 8-51, states the following:

The relationship between the objects in a workflow processing framework according to the present invention and the peripheral processes is illustrated in FIG 2. **At the bottom level are conventional platforms 20...** These conventional platforms 20 may run on any type of computer system... **A set of foundation objects 22 access the conventional platforms 20** using standardized protocols whenever possible...

The foundation objects 22 are illustrated separately from the objects 24 providing common functions in support of different applications, because in a specific enterprise environment standardized protocols may not be available and the foundation objects may need to be modified to utilize conventional platforms 20 existing in the enterprise environment or previously selected for use with the workflow processing system...

The objects 24 providing common functions in support of different applications provide “foldering” of materials... used by each case and workflow function... Business processes 26 written in conventional programming languages perform enterprise specific functions. The business processes may be written in a variety of programming languages...

The above language from Lynn et al. demonstrates that the reference is directed to a general structure or framework for software, in which the bottom level is a conventional platform, the next level is a set of foundation objects, followed by objects providing common functions, and finally business processes. The notion of developing software by determining participants and associated roles is not present, and the creation of “use cases” at the level where a participant has only one role is not present.

Rejection of Claims 12-23 under 35 U.S.C. § 102(e)

As stated above, Claims 12-23 were rejected under 35 U.S.C. § 102(e) as allegedly anticipated by Lynn et al. Applicant respectfully traverses for at least the following reasons.

Rejections under 35 U.S.C. 102(e) require that the reference (or combination of references) disclose every element of the claim. According to the MPEP, “for anticipation under 35 U.S.C. 102, the reference must teach every aspect of the claimed invention either explicitly or impliedly.” MPEP § 706.02. Lynn et al. does not refer to developing software by determining participants and associated roles, nor does it refer to use cases, or instances of the use of the software by an actor, to be used when a participant is associated with a single role.

The Official Action alleges that Lynn et al. Fig. 20, items 20, 22, and 24 amount to “a first set of participants in the use of said software” as taught by claim 1. Official Action page 2, Item 4. However, Lynn et al.’s Fig. 2 makes no mention of participants, and neither does the corresponding text, as can be verified from the excerpted section of Lynn et al., above. The present invention uses participants and various associated roles as a powerful tool in helping to envision a complex problem. Because this notion is absent from Lynn et al. and the other art of record, applicants respectfully submit that the present invention, as claimed, patentably defines over the prior art.

It is further alleged in the Official Action that Figs. 3, 4a and 4b of Lynn et al. include disclosure corresponding to Applicant’s “use case.” However, the definition of the word “case” as used in Lynn et al. does not correspond to the definition of the term “use case” in the present Application. Although Lynn et al. never directly defines the term “case,” the intended definition can be gleaned from the context. Consider the following occurrences of the term “case” in Lynn et al.:

During that interval BLCaseWorkList prefetches the next case to the client workstation of the user. This limits the idle time of the employees between and during cases and increases the output of the employees. Lynn et al. column 7 lines 62-65.

Whenever a case is retrieved from the database platform in conventional platforms 20, a prefetch of that case is completed. Lynn et al. column 8 lines 12-13.

Once a worklist has been obtained, the user can select on of the cases on which to work.

Lynn et al. column 8 lines 39-40.

From the above quotations it is clear that the reference defines “case” as an item of work or a thing to be worked upon by employees. In contradistinction, the present invention defines the term “use case” as set forth above: an instance of the use of the software by an

actor. In other words, a use case could be, for example, an employee scanning a database for specific file types, or copying sensitive and privileged company files to a new location. In this regard, use cases encompass a much broader possible range of events than the cases of Lynn et al., because they are instances of use of software, not simply files to be worked on.

Claim 1 of the present invention provides:

1. A method for developing software, the method comprising:
defining a focus area which represents:
a business process to be performed by the software under development; and
one or more first participants in said business process, at least one of said first participants having a plurality of roles;
decomposing said focus area into one or more sub-focus areas, each of said sub-focus areas including:
a subset of said business process; and
one or more second participants in said subset of said business process, each of said second participants having only a single one of said plurality of roles;
creating a use case based on a first one of said one or more sub-focus areas, said use case comprising an instance of usage, by a one of said second participants, of a first subset associated with said first one of said sub-focus areas; and
creating source code to perform acts performed in the course of providing said first subset of said business process.

(Emphasis added for clarity).

Claim elements incorporating the notion of participants, and also incorporating the notion of a use case, are similarly found in independent claims 12 and 18 of the present Application. More specifically, claims 12 and 18 teach: "said first focus area representing...a set of first participants..." and "generating a use case based on said third information..." As discussed with reference to claim 1, Lynn et al. does not show these elements in independent claims 12 and 18, and thus Lynn et al. can not be said to anticipate. Claims 13-17 and 19-23 depend from claims 12 and 18, respectively, so they too are deemed to be patentable over the cited art.

Rejection of Claims 1-11 under 35 U.S.C. § 103(a)

Claims 1-11 were rejected under 35 U.S.C § 103(a) as being allegedly obvious over Lynn et al. in view of Tsukakoshi. As with rejections under 35 U.S.C. 102(e), rejections under 35 U.S.C. 103(a) require that the reference (or combination of references) disclose every element of the claim. As stated in the MPEP, "the prior art references (or references when combined) must teach or suggest all the claim limitations." MPEP § 706.02(j).

Tsukakoshi was cited for reasons related to the generation of source code. Applicants do no contest the assertion in the Official Action, page 5, that generating source code during software development has been a general practice in the art. However, Tsukakoshi, like Lynn et al., does not include or suggest the elements of participants and use cases, as described above in connection with claim 1. Thus, because Tsukakoshi does not cure the above-identified deficiency of root reference Lynn et al. as applied to Applicant's invention, the combination of the Lynn et al. and Tsukakoshi references do not render obvious Applicant's invention.

As a result, it is respectfully submitted that independent claim 1 is patentable over the Lynn et al and Tsukakoshi references, whether taken alone or in combination. As claims 2-11 depend either directly or indirectly from claim 1, they are believed allowable for the same reasons. Withdrawal of the rejections under 35 U.S.C. § 102(e) and 35 U.S.C § 103(a) is thus earnestly solicited.

CONCLUSION

Applicant believes that the present reply is responsive to each of the points raised by the Examiner in the Office Action, and submits that Claims 1-23 of the application are in

DOCKET NO.: BB006 (USYS-0099)
Application No.: 09/834,357
Office Action Dated: September 16, 2003

PATENT

condition for allowance. Favorable consideration and passage to issue of the application at the Examiner's earliest convenience is earnestly solicited.

Date: January 9, 2004



Nathaniel A. Long
Registration No. 53,233

Woodcock Washburn LLP
One Liberty Place - 46th Floor
Philadelphia PA 19103
Telephone: (215) 568-3100
Facsimile: (215) 568-3439